# BlackBerry Wireless Handheld User Interface Style Guide

Version 1.0

BlackBerry Java Development Environment Version 1.0
BlackBerry Wireless Handheld User Interface Style Guide
Last modified: 20 November 2003

Part number: PDF-06946-001

While every effort has been made to achieve technical accuracy, information in this document is subject to change without notice and does not represent a commitment on the part of Research In Motion Limited, or any of its subsidiaries, affiliates, agents, licensors, or resellers. There are no warranties, express or implied, with respect to the content of this document.

Research In Motion Limited
295 Phillip Street
Waterloo, ON N2L 3W8
Canada

Research In Motion UK Limited
Centrum House, 36 Station Road
Egham, Surrey TW20 9LF
United Kingdom

Published in Canada

# Contents

1

# User interface design principles

This section provides information on the following topics:

- Overview
- User interface design principles

# Overview

This document provides guidelines for designing a user interface for your BlackBerry handheld application. It contains descriptions of BlackBerry controls, as well as guidelines and suggestions for their usage.

## Related resources

Refer to the following resources for additional information:

| Resource | Description |
|---|---|
| *BlackBerry Java Developer Guide* | The *BlackBerry Java Developer Guide* provides an introduction to developing applications for the BlackBerry Wireless Handheld. It contains sample applications as well as conceptual material about the BlackBerry Java APIs. |

# User interface design principles

An effective user interface is extremely important to an end-user application. Designing an application according to specific design principles can simplify a complex application. A simple, effectively designed user interface creates a good user experience and dramatically increases user productivity.

Consider the following principles when designing an application user interface.

## User control

The user should always be in control of the application; to achieve user control, keep in mind the following principles:

### Responsiveness

The user must be able to navigate and perform actions quickly without having to wait for the application. If there are long delays between the time the user performs an action and the processing of the action, the user will not feel in control. Responsiveness can be improved by writing efficient source code where possible, or by prompting the user with an hourglass icon, informing the user that the application is processing.

User interface response time should be, at worst, 200 ms (1/5 of a second) or better. If an operation is expected to take a long time (for example, a text search or a database query), then it should be executed in the background. If this is not possible, a splash screen or wait icon should be used to reassure the user that an error has not occurred.

### Customization

Because user preferences differ, users should have the ability to customize the interface of the application. This includes deciding what information to display in the message list and the order in which to display information. By customizing the interface, the user can create an application that best suits their typical, day to day usage pattern. However, the application should provide defaults that are helpful for the majority of users.

### Perform tasks at hand

The best interfaces are the ones that make tasks easier for users, without drawing attention to the interface. The user interface should be designed to allow the user to simply and intuitively perform a task. The best interfaces are the ones that are hardly noticed.

## Intuitiveness

An interface should be as intuitive and straightforward as possible. One of the best ways to achieve this is by using the object-action paradigm. The object-action paradigm refers to a design principle that associates specific objects in the application with the actions that are performed on them. For instance, on the BlackBerry handheld, a user can select a message from the message list and then perform an action on it (such as view, reply, forward, or delete).

## Context

Along with its intuitive design the BlackBerry handheld excels in its ability to provide a sense of context to user interface design. With respect to a BlackBerry handheld application, context describes the circumstances under which an action, or a particular situation, occurs. Context infers that the user interface is adaptable and flexible, and changes to best meet the needs of the users depending on the way in which the interface is used. In BlackBerry handheld applications, menus, menu options, and menu defaults change depending on the context in which the application is used. For example, when a user composes a message, the menu options change to provide different functionality depending on the position of the cursor. When the cursor is positioned in the **To** field, the menu options change to display only those options relating to this action.

Providing effective context can prove very difficult since it requires a great deal of planning and usability testing. It is, however, extremely important to the success of your application since a good sense of context can create an extremely intuitive and easy to use application.

## Consistency

Consistency should exist throughout each task and between applications. Consistency improves the user experience by providing a familiar set of actions or tasks across all applications on the handheld. By creating this familiarity, the chances that the user will become confused or “lost” are greatly decreased. The onus is on you, as a developer, to be aware of other applications being written for the handheld so that your application interface is a consistent part of the whole BlackBerry handheld solution. Consistency enables common interface elements to be reused.

In addition to providing consistent individual operations, block operations should be performed as though they were individual operations. This is especially important if an operation fails in the middle (for example, pasting data into a nearly full buffer or deleting files from a folder).

## Clarity

The interface should be clear, not only from a visual point of view, but also from a conceptual point of view. Clarity is an especially important design aspect because the BlackBerry handheld has a small viewing screen. If the interface causes distractions by breaking visual flow, users might have trouble maintaining control. If you are using icons, keep in mind that they will be small and if they are cluttered, it will not be clear to the user how the icon is related to the interface that contains it.

BlackBerry handheld applications should not be considered Windows® applications. Using too many controls will only emphasize the small size of the display.

Dialog and status boxes should contain brief and concise text. This text should not merely indicate to users what they have done incorrectly; they should also indicate what action is necessary to correct the problem.

## Aesthetics

The BlackBerry handheld is a two-way wireless handheld — it is not a desktop solution. As such, BlackBerry handheld applications should not be compared with Microsoft® Windows® applications. Three-dimensional graphics, sounds, and flashing cursors on the handheld display are inefficient and distracting. A simple design is generally the way to make handheld applications aesthetically pleasing.

When the information on the screen changes, limit the changing data to one item at a time, by keeping the user focused on one place at a time. When presenting a lot of information to users, display the most important information first. For example, when presenting each address book entry, display the name fields first, followed by the email address, phone, fax, and pager numbers fields, and leave the Notes field to the end.

In addition, the application should not cause any unnecessary screen repaints, nor should it take focus from other applications without permission.

The BlackBerry user interface engine maintains an aesthetically pleasing appearance by taking advantage of the vertical nature of the trackwheel. Menu items and lists are presented in a vertical fashion. Double-clicking (as opposed to two quick single clicks) is not desirable based on the size of the handheld and the size of the trackwheel itself.

## Feedback

Feedback describes the application's response to user input (for example, the appearance of a menu when a user clicks the trackwheel). Feedback is crucial to the success of an application since it defines and shapes the very way in which a user interacts with the BlackBerry handheld. Prompt feedback is essential and the application should respond quickly to the commands of users. For instance, one of the learning curves for new users of the BlackBerry is typing on the keyboard. Since keyboard typing may take a while to get used to, new users will become frustrated quickly if feedback from typing or invoking screens or menus is slow.

## Forgiveness

The interface must allow users to change their mind and undo commands; essentially, it must be forgiving. The following principles describe user interface forgiveness as it relates to the BlackBerry handheld.

First, users should not be enabled to perform any destructive tasks without being warned (for example, an action such as deleting a message must be confirmed before the action can occur). When a destructive action occurs (or is about to occur) the user is always prompted by a warning dialog box. This warning enables the user to undo or reverse an action before it is too late. However, be careful with over prompting; it can cause a user to routinely ignore all prompts and therefore cause unwanted, irreversible actions.

Second, avoid placing common menu items close to others that undo the task at hand (for example, the Cancel menu item is not close to the Hide menu item).

## Screen title

Every screen will have a title on the first line of the display. The title provides cohesion between screens in the application. The title line should be used not only to inform users where they are, but also to display brief information to help the users with the particular task at hand. For example, when users perform a search in the address book application, the first line of the screen will display the screen title plus the text typed to invoke the search.

## Controls

Controls are tools such as icons and buttons that the user selects to issue commands. Keep the number of controls on the screen display at any one time as small as possible. Too many controls on the screen display will only emphasize the small size of the screen.

## Errors

When encountering error conditions, the user should be informed of the error without being inconvenienced. Attempts should be made to make the failure “graceful” such that no information is lost. For example, if inserting a character in the middle of a full field, prevent further input (and display a brief error message) rather then throwing out characters at the end of the buffer without warning.

2

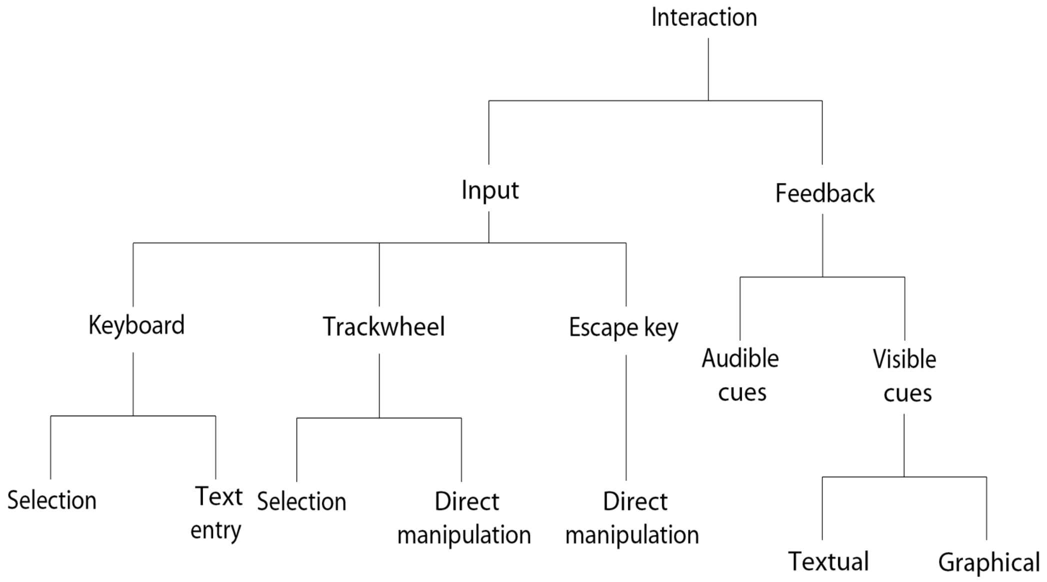
# BlackBerry interaction model

This section provides information on the following topic:

- BlackBerry handheld interaction mechanisms
- Navigating in a BlackBerry application

# BlackBerry handheld interaction mechanisms

When designing a user interface for your BlackBerry application, consider the BlackBerry input and output mechanisms. The following diagram describes the input mechanisms for the BlackBerry Wireless Handheld.



**BlackBerry input mechanisms**

While the BlackBerry keyboard and trackwheel are easy to use, they differ greatly from common input devices such as the standard desktop keyboard and mouse. You should take into account the following considerations when designing your user interface.

## Keyboard and trackwheel interaction

The user should not have to alternate between using the keyboard and using the trackwheel to navigate. For example, if users are entering data using the keyboard, they should not have to use the trackwheel for some form of input and then go back to using the keyboard.

Most actions should be performed using either the keyboard and the trackwheel.

### Trackwheel

The trackwheel is unique in that it has dual input capabilities since it can be clicked and rolled by the user. While simple to operate, the wheel also has drawbacks. Quite often, users will overshoot with the trackwheel or accidentally click it while rolling.

To counter this, the following should be considered:

- forgiveness principles should take this into account
- menu items and dialogs should not be grouped in such a way to cause the user to accidentally do something destructive.

### Escape button

The Escape button is used to cancel operations such as selecting text or sending a message. The Escape button should always return the user to the previous home screen.

### Feedback

Feedback is accomplished on the BlackBerry handheld by both visible and audible means. The BlackBerry is able to use an assortment of audible tones to signal alarms, appointments, or incoming calls. It also displays dialogs or text messages to display information to the user.

## Navigating in a BlackBerry application

The main method of navigation in a BlackBerry application is through the use of the drop-down menu. The menu provides the user with a list of actions that can be performed on a given item. A user clicks the trackwheel to access the menu. The menu appears in the top right corner of the screen.

You should avoid using buttons or cluttering the screen with custom controls. The menu provides the means to display actions and tasks while using up little screen space.

Use default navigation whenever possible. When the user clicks the Close menu item or presses the **Escape** button, the application should return to the previous screen or, if there is no previous screen, the application should close. If the user has entered data or modified options on the screen, a confirmation prompt should appear to enable users to save or discard changes.

Clicking the trackwheel should always lead to the next natural action, which is typically a return to the previous screen or close the application.

When a user exits an application, the handheld Home screen should appear.

The following lists additional guidelines for trackwheel and keyboard interaction:

- Remapped keys must always be returned to their original state upon exiting the application.

- In multi-line text entry mode, the **Enter** key enables the user to input a new line.
- The application must support text and numerical characters for data input based on the QWERTY keyboard, including text modifier keys when appropriate.
- In text entry mode, the trackwheel will move the cursor appropriately: moving the trackwheel up or down moves the cursor up or down, pressing the **ALT** key and moving the trackwheel up or down moves the cursor left or right.
- In numeric entry fields, verify that the keyboard automatically defaults to numeric entry instead of text. The user should be able to input numbers in a numeric entry field without having to press the **Alt** key to enter numbers instead of text.
- In an application, pressing the **Escape** button enables the user to return to the Home screen. Pressing the Escape button repeatedly should always return the user to the Home Screen except from within a text entry field.
- From within the application, the Cancel or Close option should always "cancel" the operation. The application should move one screen back toward exiting the application and returning to the Home Screen.
- All icons that require a hot key (such as an icon on the Home Screen) will have a unique hot key. Each hot key that is used for any icon shortcut is unique to that icon.

## BlackBerry user interface checklist:

- Pressing the trackwheel Always invokes a menu
- Pressing the Escape button Always returns to a prior screen or exits the application
- The main application screen Always includes an Exit action as the last menu item (for example, Close or Exit)
- The user must be prompted to save when leaving a screen in which the data has been modified
- Always prompt a user with a warning dialog before data is destroyed
- All icons on the home screen must be enabled with a unique hot key
- Verify that the application obtains and loses focus appropriately. The application should not execute while it does not have the focus.
- Verify that the keyboard is mapped correctly when inputting text into a field and remapped correctly when the application exits
- The user should always return to the Home screen when an application exits
- BlackBerry applications should always be written to support localization. Text strings should be retrieved from a separate resource bundle. Numbers, dates and time formats should reflect the locale of the user.

3

# Designing the interface

This section provides information on the following topics:

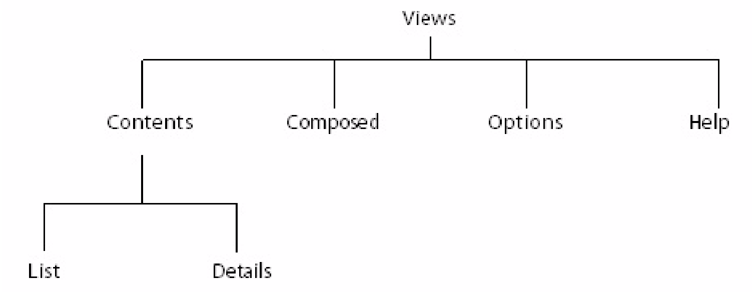
- BlackBerry views
- User interface elements
- Creating an effective main screen
- Browser design principles
- Handheld error prevention and recovery

# BlackBerry views

The BlackBerry user interface was designed with simplicity and efficiency in mind. This was accomplished by designing an interface that allows users to easily perform tasks and access information

Conceptually, the BlackBerry user interface can be broken down into a set of standard view types. A view type defines the basic layout of the application screen and application data.

The BlackBerry user interface consist of the following view types:



**BlackBerry view types**

The views are task oriented, since each view type was created to accomplish a single, specific task. For example, the Composed screen enables users to compose notes or messages, while the options screen enables them to view or modify an applications Options.

## Contents view

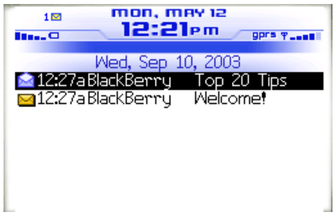
The contents view, which is also called the functions view, typically forms the main screen of the BlackBerry application. It is displayed when an application is selected from the Home screen and itlists the components or objects associated with the application. The components can be ordered or unordered. The meaning of an object is not necessarily affected by its order within the list.

The contents view type has two sub-types:

- list view
- details view

## List view

The list view lists the objects within the applications, usually alphabetically. For example, the message list is a type of list view. The message list lists all messages that were sent or received by the user. The list displays the name, subject, time, and date of the message, as well as an icon that displays whether or not the message was sent or received.



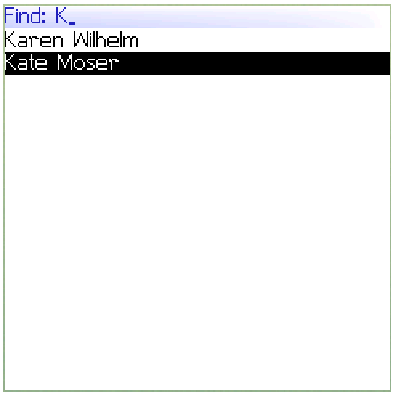
Wed, Sep 10, 2003

12:27a BlackBerry Top 20 Tips

12:27a BlackBerry Welcome!

**Message list**

Since the list view often contains many items, the Search functionality should be made available to allow the user to search for and access information given a text string or pattern. In the Message list screen, the Search function can be accessed from the menu. In the address book, which is also a list view, the Find function is used to perform a search on objects within the list.



Find: K_

Karen Wilhelm

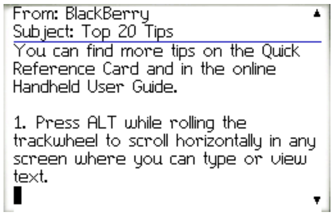
Kate Moser

**Address book**

The user enters the first letter of the name of the contact that they are searching for. The list narrows to contain only contacts with names that start with that letter. As the user enters more letters from the contact's name, the list narrows to contain only the names of contacts that match the search string.

## Details view

The details view is typically accessed from a list view screen. The details view provides another layer of information about the current object in the list. The type of information displayed depends on the type of object that was accessed. For example, when a user clicks the trackwheel on a message in the message list, an email displays.



From: BlackBerry
Subject: Top 20 Tips

You can find more tips on the Quick Reference Card and in the online Handheld User Guide.

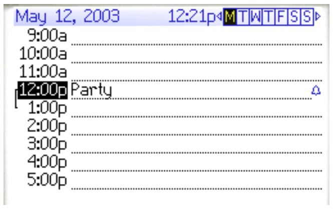
1. Press ALT while rolling the trackwheel to scroll horizontally in any screen where you can type or view text.

**BlackBerry message**

The details view type displays information in greater detail, and in some cases, allows you to modify the data. The address book and task applications provide other examples of details view types.

## Composed view

The composed view is typically accessed by clicking an icon from the home screen. The composed view differs from the details view in that it represents a view of an object in which relationships of the parts contribute to the overall meaning of the application. In other words, the composed view contains fields that allow you to interact directly with the application. For example, when the user clicks on the calendar application in the home screen, the current week, month or day is displayed in a composed view type.



May 12, 2003 12:21p

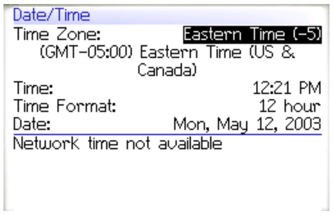
9:00a
10:00a
11:00a
12:00p Party
1:00p
2:00p
3:00p
4:00p
5:00p

**BlackBerry calendar entry**

You can view or edit each field in the week view calendar format. Editing data at the field level in this view is reflected in the overall calendar application.

## Options view

The options view displays information about the characteristics, attributes, or properties of an object. It enables the user to view and modify the options or settings of a particular application. Not all options can be modified.



Date/Time

Time Zone: Eastern Time (-5)

(GMT-05:00) Eastern Time (US & Canada)

Time: 12:21 PM

Time Format: 12 hour

Date: Mon, May 12, 2003

Network time not available

**Date / Time screen**

The Date/Time screen enables users to change the way in which the date and time information display on the handheld. An options view type is typically provided for each type of object.

## Help view

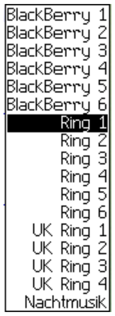
The help view is not yet implemented.

# User interface elements

User interface (UI) elements, such as choice fields and dialog boxes, make up the core of a BlackBerry application. They represent the lowest level at which style can be applied to an application. Their adherence to a proper style is critical since they represent the exact point at which a user interacts with the application.

## Choice fields

Choice fields provide users with a horizontal list of options from which they can scroll up or down to choose a value. A value is selected by rolling the trackwheel and clicking the selected value.



BlackBerry 1
BlackBerry 2
BlackBerry 3
BlackBerry 4
BlackBerry 5
BlackBerry 6
Ring 1
Ring 2
Ring 3
Ring 4
Ring 5
Ring 6
UK Ring 1
UK Ring 2
UK Ring 3
UK Ring 4
Nachtmusik

**Ring tone choice field**

Choice fields should order choices so that the index in the choice is directly correlated to the collation or sort order, if applicable. The user will expect that rolling in the “up” direction will increase this value, where “up” is defined as a UI constant, usually down. So the default behaviour is that up is down and down is up. This should be changed through the constant Ui.UP.

For example, suppose there is a choice of numbers: 0, 1, 10, 100. The choices should be choice[0] = 0, choice[1] = 1, choice[2] = 10, choice[3] = 100.

As another example, suppose the list is the months of the year. The choices should be choice[0] = JANUARY, choice[1] = FEBUARY, ..., choice[11] = DECEMBER.

Choice field entries should be ordered logically, usually according to normal order, size or alphabetically. For example, JANUARY, FEBRUARY, MARCH or BLUE, GREEN, RED.

**Tip:** The choice field is described in the BlackBerry Java API by the `ChoiceField` class. The ChoiceField. class contains the functionality for creating customizable choice field components. It contains methods that enable you to define the size of the choice field as well as the items that appear in the field.

for more information about the ChoiceField class refer to the *BlackBerry Java API Javadocs*

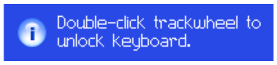
## Notification dialog boxes

Notification dialog boxes should prompt a user whenever an operation occurs or when a consequential action is about to take place such as deleting an item or changing the handheld time or date. Prompting with a dialog box provides the forgiveness that protects users from themselves.

The BlackBerry handheld provides the following dialog box types:

### Information

The information dialog provides information about the results of a recently executed command. It is a modal dialog that offers the user no choices. It acts solely as a way to prompt the user with important information.



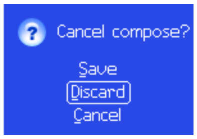
Double-click trackwheel to unlock keyboard.

**Information dialog box**

The user must acknowledge the message by clicking the OK button.

### Question

The question dialog box prompts the user whenever a consequential action is about to occur. For example, the user is prompted when a message is about to be deleted. The user must provide input before the operation will proceed. The wording of the message must always be in the form of a question. For example, “Save Changes to ...?” or “Cancel Compose?”



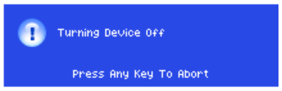
Cancel compose?

**Question dialog box**

The button text must always describe the action that the user is about to take and must always answer the question.

### Exclamation

The exclamation dialog box warns the user that a critical action or in some cases an error has occurred. The exclamation dialog box also appears when the handheld is turned off.

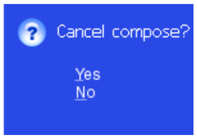


Turning Device Off

Press Any Key To Abort

**Exclamation dialog box**

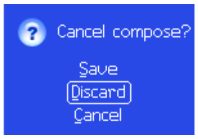
The wording of dialog questions should be consistent across the application. Yes/No dialogs boxes, or binary dialog boxes that contain only the Yes and No options, should be avoided if possible.



Cancel compose?

**Bad: Do you want to save the changes?**

If possible, the user should be provided with another option and more information on the outcome of the event. Dialog boxes should be simple and direct but verbose enough to avoid confusion.

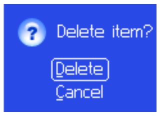


Cancel compose?

**Good: Changes have been made.**

Cancel dialog boxes should always provide an escape route or cancellation action.

In accordance with the principal of non-destructiveness, the Delete item should always have a confirmation dialog box in case the user meant to edit the item.



Delete item?

**Proper BlackBerry confirmation dialog box**

The application should not inhibit the use of notification dialog boxes. That is, when a dialog has the focus, no other application can obtain focus and execute.

## Menus

Menus provide the user with a non-scrolling list of options from which to choose. The menu appears in the top right corner of the screen and is accessed by clicking the trackwheel. Once a menu item is highlighted (by scrolling with the trackwheel), the item can be selected (by clicking the trackwheel). Subsequently, a screen is displayed. Depending on the functionality of the item selected, this screen may be the same as the previous one.

The vertical placement of the menu options plays a very important role in designing an effective application. To increase familiarity with applications and to shorten the user's learning curve, the order of items in menus should be as consistent as possible between all applications.

Care should be taken when choosing the default item in a menu. The default item should, of course, be the item that will most likely be chosen by the user. This will also depend on the context in which the current screen was entered and the item was clicked. Choosing an appropriate default menu item will save the user time.

The default menu item will be highlighted and should appear in the middle of the list of items when the menu first appears. The menu items that are selected more often by users should immediately surround the default item, while the items that are selected less often should be placed further away from the default item. Placing common menu options close to the default menu item minimizes trackwheel scrolling by the user to locate the desired menu item.

Provide short but descriptive names for menu items. Menu item names should always be verbs (actions). Limit menu item names to 12 characters. Capitalize the first letter of each word in the menu item name.

**Note:** Be careful not to include too many items in a single menu. If your menu becomes too large, break the menu down into two or more categories and create a new menu.

The following sections describe the recommended method for determining the order and layout of menu items within a menu.

### Category ordering

A menu can be broken down into a series of functional categories. Each category contains a group of one or more related items that represent the actions to be performed. Categories are distinguished from each other by menu separators.

### Description of categories

**Continuation actions**

In some cases, an action will take several steps to complete. For example, to select an address when composing an email, a user needs to first select the address, then select the addressing method. The actions that complete the second step are called “Continuation Actions”. **Use Email**, **Use PIN**, and **Continue** when composing a message are examples of this type of action.

### View changing actions

These actions change how the current screen appears. The only example in the current code is the View All menu item which will restore a filtered list to the full list.

### Item actions

In a list screen, item actions relate to the selected item in the list. For example, in the address book, any actions related to contacts are in this category (for example, New Address, Edit Address, or Delete Address). In a record screen, the item in question is the open record. In this context, examples of item actions are Save Item or Delete Item.

### Content actions

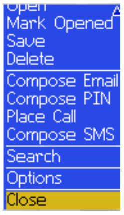
In a record screen, the lines represent content within the item that is opened. Content actions are actions related to the content within the item. For example, “Select Name” to select a name to put in a Name field in a search, “Clear Field”, or “Show Qualified Address” are examples of actions on the content of an item.

### Related function actions

These actions launch some other related function, for example, Compose Message, Search Message, and Options.

### Exit actions

Exit actions are those that will exit the current screen. These actions could be as straightforward as Cancel and Close or more subtle like Previous Message and Next Message.



Proper BlackBerry menu

## Item ordering

Within each category, actions are ordered from least to most destructive — new (creates), save (preserves), and delete (destroys).

A menu separator should be used only to separate actions in different categories.

It is important when designing the layout of your menus to consider context. Think about the cursor position and the operations that might be executed when the user clicks the trackwheel. Limiting the available menu options and displaying only those menu options which are relevant to the current context can save the user a great deal of time and lessen frustration.

**Tip:** The menu is described in the BlackBerry Java API by the `Menu` class. The `menu` class defines the functionality that is common to all menus, such as adding and removing menu items, and provides the framework for creating menus for your application.

for more information about the Menu object, refer to the *BlackBerry Java API Javadocs*

## Controls

Controls are tools such as icons and buttons that the user selects to invoke commands.

Use the least number of controls on the screen as possible. Too many controls on the display emphasizes the small size of the screen. Buttons are redundant since every click in a BlackBerry application opens a menu.

## Displaying a list of options

A list of options allows the user to provide input to the application by presenting a fixed group of choices. Depending on the circumstances, a list should be displayed to the user in one of the following ways:

- Use check boxes when the user can select multiple options.
- Use a drop-down list or radio buttons when the user can select only one option.
- For presenting numeric options, use a drop-down list when the range of numbers is less than 20.
- For larger ranges of numbers, use a gauge field (progress bar).

Be sure to provide a reasonable default value for each option list.

## Forms

Forms enable a user to enter large strings of data into the application. When implemented correctly, forms provide an effective means to enter text using the BlackBerry handheld. Consider the following principles when designing a form.

When possible cache any data that has already been entered into a form and reload the information into the form the next time the page is loaded. This saves a user from re-entering the same data each time the page is loaded, such as a username.

To prevent input errors, you should clearly identify the required fields in the UI, and verify that the required fields are filled in correctly before submitting the form. If the form has errors, provide a descriptive message to the user describing which field needs to be corrected.

When a field has no external dependencies and contains only simple information, then it might be possible to validate the field either at the time of information entry or when the user leaves the field. For example, in the BlackBerry calendar application, the date field automatically corrects apr 31 to apr 30 when April is selected from the Month list. For fields that depend on other fields or those that require complex input (for example, phone numbers) validate the form when it is submitted.

## Fields

Fields enable users to enter formatted strings of text into an application. Consider the following principles when designing a form.

Use a text filter on a field to permit only valid characters (for example, for an email address or phone number field), and set an appropriate maximum number of characters.

When creating a password field, be sure to echo the text to the screen as an asterisk. This helps to shield the password from third-party eavesdroppers.

Non-editable fields should be distinguishable from editable fields. It should be obvious to users when they cannot edit a field.

The BlackBerry Wireless Handheld supports the following field types:

| Field type | Description |
|---|---|
| BasicEditField | an editable simple text field with no formatting; this is the most common field type and should be used to display simple text |
| AutoTextEditField | an editable text field designed to provide autotext support; when a word is entered that is in the autotext database for the current locale, this field replaces it |
| BitmapField | displays a bitmap |
| ButtonField | displays a button control |
| CheckBoxField | displays a checkbox control; the checkbox field toggles a value<br>pressing the space key, or clicking the trackwheel (when a checkbox field has the focus) toggles its state |
| ChoiceField | displays a list of scrolling choices |
| DateField | displays a date; the date can be displayed in a number of different formats |
| EditField | displays a simple text field that can be edited by the user |
| GaugeField | displays a horizontal bar that you can use for numeric selection or as a progress indicator; the GaugeField provides a level of responsiveness to the application |
| LabelField | displays a simple label |
| ListField | contains rows of selectable list items |
| NullField | represents a field of 0 size |
| NumericChoiceField | displays a field that supports choosing from a range of numeric values |
| ObjectChoiceField | displays a choice field that supports a list of object choices |

| Field type | Description |
|---|---|
| ObjectListField | displays a list of objects |
| PasswordEditField | displays an editable text field designed specifically to handle password input.<br>When the user enters text into a PasswordEditField, the text is stored in a private buffer and appears to the user as a string of asterisks. This field does not have any automatic input replacement nor does it support cut and paste operations, so the typed text is exactly what is typed. |
| RadioButtonField | displays a radio button control |
| RadioButtonGroup | displays a set of related radio button fields |
| RichTextField | displays a read-only field that supports richly formatted text; this field does not permit the user to enter or edit its text |
| SeparatorField | displays a field which draws a horizontal line across it's width. The SeparatorField is used to separate groups of choices within menus |
| TreeField | displays a simple field to show a tree structure to show a relationship between items (such as folders and documents) |

**Tip:** for more information about Field types, refer to the *BlackBerry Java API Javadocs*

Lists enable you to create directories of items through which users can scroll and select individual or multiple entries. Lists are scrolled vertically using the trackwheel. The BlackBerry address book is an example of a `List` object.

## Ribbon icons

Icons should be simple to comprehend and should represent a meaningful association with the application it calls. Icons must be 26 by 26 pixels in size, with a color depth of 65,000 colors. When creating an application, include both a color icon and a monochrome icon in the resource files for the application. At start-up, the application will automatically select the appropriate icon to display depending on whether or not the handheld has a color screen.

# Creating an effective main screen

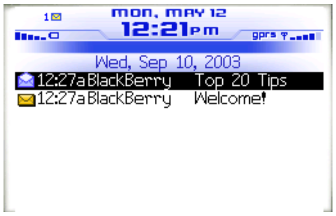
Most BlackBerry applications share similar main screen characteristics in terms of appearance and functionality. These characteristics are dictated both by the design of the trackwheel and the size and shape of the screen. For this reason, most BlackBerry applications should adhere to a common set of standards.

The main screen of a typical BlackBerry application should have the following characteristics:

- title bar at the top with a separator line after the title

- optional standards sized non-scrolling region at the top of the screen (called the banner) that displays the date and time and the radio and battery status of the handheld. Note that while the banner is optional and is not suited for all applications, it is a good idea to include at least battery and signal information, especially in applications that use the radio.
- main scrolling section with vertical layout
- menu with a Close menu item
- close action when the user clicks the Close menu item or presses the **Escape** button

Each screen displays an appropriate title bar that describes the contents of the screen. For game applications, an appropriate title bar is required only when the user initially enters the game from the Home screen.



1

mon, MAY 12

12:21pm

gprs

Wed, Sep 10, 2003

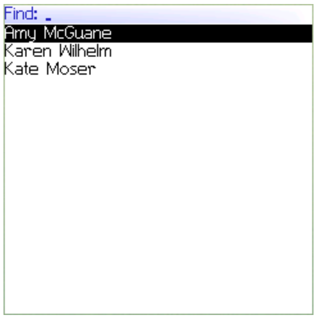
12:27a BlackBerry Top 20 Tips

12:27a BlackBerry Welcome!

**Message List main screen**

In the previous figure, the Message list screen of the mail application adheres to the principles stated previously. It features a non-scrolling region containing the date and time, a vertically scrolled message and a menu with a close item. The application closes when the user presses the **Escape** button.

In the following figure, the search functionality of the address book uses the title bar to prompt the user to enter a search string.



Find: _

Amy McGuane

Karen Wilhelm

Kate Moser

**Search screen**

**Tip:** The application main screen is described in the BlackBerry Java API by the `MainScreen` class. The `MainScreen` class defines the components, such as the title bar and main scrolling regions, that make up the common features of a typical BlackBerry application. Most of the main screen functionality is provided in this class.

for more information, refer to the *BlackBerry Java API Javadocs*

# Browser design principles

In a wireless environment, you should typically deliver related content on as few pages as possible. Although a page with more content might take a few seconds longer to download to the browser, users do not have to make subsequent requests, and the information is available even when users move outside an area of wireless coverage. Because the BlackBerry handheld provides a trackwheel for scrolling, users can scroll through several screens of text easily.

For example, in WML pages, put related cards in the same deck whenever possible so that the document only has to be loaded once. If the deck contains a relatively large card that many users might not want to view, save the card in its own deck to minimize download time.

If you divide related content into more than one page, make links to related content easily accessible. Make sure that links to related content are visible in a non-scrolling area of the page or at the top of the page.

Design web pages to use the BlackBerry handheld screen effectively. The handheld has a larger screen than many other mobile devices, such as cellular phones.

## Text entry

Text entry on a cellular phone keypad can be difficult, so effective site design for WAP browsers often mandates that, whenever possible, users should not be required to type text using the keypad.

This restriction is not applicable to the browser; users are often comfortable using the keyboard to type text into web forms. You should not require users to type redundant or anticipated information, but you can expect users to type text using the keyboard when appropriate.

The browser supports both the `<input type="text">` and `<textarea>` elements in HTML, and `<inputtype="text">` in WML.

## Minimize download time

Download time is affected by three factors: content size, wireless network, and protocol characteristics. For example, a 15 KB file can take between 30 and 50 seconds to download through a WAP gateway on a GPRS network.

The best way to improve download time and provide a high-quality user experience is to reduce the size of your web pages. Do not include unnecessary content in web pages. Use images only when necessary, and reduce image file size as much as possible.

Consider using a cascading style sheet (CSS) to render your HTML page. A cascading style sheet enables you to define the look of your web page by defining styles and fonts externally. Cascading style sheets dramatically decrease the time it takes to download and display your webpage.

## Improve rendering time

Rendering on the browser is much faster than download times, but large content can still require several seconds to parse and display. WML typically renders more quickly than HTML or XHTML content.

The Mobile Data Service (MDS) optimizes browsing by processing HTML content before sending it to the browser. The Mobile Data Service filters unsupported markup and converts content into a tokenized format that the browser can display efficiently.

In addition, you can improve rendering times by processing and filtering content at an intermediate server between the web server and the handheld.

## Images

- use well-defined images and clear color contrasts
- fonts that are saved as images should not be anti-aliased; anti-aliasing smooths edges by blending the background and foreground colors and anti-aliased images do not display optimally
- if you need to shrink an image to better fit the smaller screen, when possible, you should redraw the image; scaling down the image results in blurred edges that display poorly
- in web pages, remember that users can specify whether images should be loaded or not; images should not be critical to the effectiveness and usefulness of your web site

**Note:** Remember that not all users have the same handhelds. Create images that render well on both monochromatic and color screens. Always provide descriptive alternate text for images just in case.

## Web content

To deliver the appropriate content to wireless browsers, you can write a script to deliver different content depending on two factors:

- **Browser type:** In an HTTP request, the browser includes a User-Agent header that identifies itself. Use this header to deliver different content to desktop browsers than to wireless browsers. You can also deliver different content to BlackBerry handhelds than to other devices.

- **Content type:** In an HTTP request, the browser includes an Accept header that specifies the content types that it accepts. Use this header to deliver content that the browser can accept. For example, the browser on Java-based BlackBerry handhelds supports WML and HTML, but the browser on C++-based BlackBerry handhelds supports WML only; handhelds with color display support .jpg images, but other handhelds do not.

# Handheld error prevention and recovery

Applications must warn the user about data loss before executing any function that might result in data loss.

For any text entry field where data has been entered, the application must post a message box warning the user of data loss for any functions that can result in data loss (for example, cancel or close).

It is important to test whether or not the application will continue to function properly when it loses focus, is halted, or is sent to the background.

Perform the following actions to test for proper functioning:

1. Press and hold the Alt key and press the Esc button to task switch.
2. Insert the handheld into the holster.
3. When an email or calendar notification occurs, remove the handheld from the holster.

   If the application is functioning correctly, it will continue to run in the background or maintain state until it regains the focus.